1M

УДК 004.434

В.І. Межуєв
Бердянський державний педагогічний університет, Україна
Україна, 71100, м. Бердянськ, вул. Шмідта, 4

Інформаційна технологія предметно-орієнтованого математичного моделювання

V.I. Mezhuyev
Berdyansk State Pedagogical University, Ukraine
Ukraine, 71100, Berdyansk, Schmidt str., 4

Information Technology of Domain-Specific Mathematical Modelling

В.И. Межуев
Бердянский государственный педагогический университет, Украина
Украина, 71100, г. Бердянск, ул. Шмидта, 4

Информационная технология предметно-ориентированного математического моделирования

У статті здійснений аналіз інформаційної технології (ІТ) предметно-орієнтованого моделювання (DSM), сутність якої полягає у створенні та застосуванні специфічних для предметних областей мов моделювання. Доведено, що незважаючи на потужність ІТ DSM, її теоретична основа та практична реалізація має ряд недоліків. Запропонована інформаційна технологія предметно-орієнтованого математичного моделювання, що дозволяє подолати недоліки ІТ DSM.
Ключові слова: предметно-орієнтоване моделювання, предметно-орієнтоване математичне моделювання.

The analysis of the information technology (IT) of Domain-Specific Modelling (DSM) is given, the essence of which is development and application of the specific for domains modelling languages. It is proved, that in spite of the power of IT DSM, its theoretical base and practical implementation have some disadvantages. The information technology of domain-specific mathematical modelling is proposed, which allows to overcome the weaknesses of IT DSM.
Key words: domain-specific modelling, domain-specific mathematical modelling.

В статье проведен анализ информационной технологии (ИТ) предметно-ориентированного моделирования (DSM), сущность которой состоит в создании и применении специфических для предметных областей языков моделирования. Доказано, что несмотря на мощность ИТ DSM, ее теоретическая основа и практическая реализация имеет ряд недостатков. Предложена информационная технология предметно-ориентированного математического моделирования, которая позволяет преодолеть недостатки ИТ DSM.
Ключевые слова: предметно-ориентированное моделирование, предметно-ориентированное математическое моделирование.

Вступ

У сучасній системній інженерії наразі можна виділити дві основні технології комп'ютерного моделювання предметних областей (ПрО):

1 General-Purpose Modeling (GPM) - моделювання загального призначення. До GPM відносяться такі мови моделювання, як: UML (Unified Modelling Language), SysML

(Systems Modelling Language), нотації методологій IDEF (Icam DEFinition), SSADM (Structured Systems Analysis and Design Method) та ін., що дозволяють відтворювати аспекти різних ПрО, незалежно від їх природи. Переваги та недоліки цього підходу були розглянуті нами у попередніх роботах, зокрема на прикладі уніфікованої мови моделювання UML [1].

2 Domain-Specific Modeling (DSM) - предметно-орієнтоване моделювання (або моделювання, що орієнтоване на предметні області).

Метою даної роботи є розкриття сутності інформаційної технології (IT) предметно-орієнтованого моделювання для виявлення переваг та недоліків цього підходу. На основі здійсненого аналізу визначаються принципи інформаційної технології предметно- орієнтованого математичного моделювання, яка дозволяє подолати недоліки IT DSM.

Аналіз інформаційної технології предметно-орієнтованого моделювання

IT GPM є застосовною до різних предметних областей, тоді як поняття DSM є специфічними для конкретної ПрО. Сутність IT DSM полягає у створенні мов моделювання, що орієнтовані на розв'язання задач певної ПрО. Такі мови називають предметно-орієнтованими мовами (англ. Domain Specific Language, DSL), на відміну від мов моделювання загального призначення.

Зазначимо відмінність DSL від мов програмування. Вона полягає у тому, що мова програмування будується в поняттях: розгалуження, цикл, лінійна послідовність операторів та ін., а для підтримки ПрО використовуються бібліотеки програмних функцій. В DSM для моделювання застосовують специфічні для ПрО поняття. Наприклад, DSL для конструювання одягу буде включати поняття: викрійка, лінія розрізу, шов, борт, комір й ін. Таким чином, сутність IT DSM полягає в розробці моделей ПрО у поняттях, близьких до семантики ПрО, а не деяких універсальних концепціях, зокрема даних й алгоритмів.

Чому для моделювання ПрО необхідно створювати окрему DSL, коли можна використати існуючу мову моделювання загального призначення та стандартний CASE (Computer-Aided Software Engineering) засіб? У той час, як цілком можливим є застосування «стандартної» технології, виділяють наступні переваги використання DSM у моделюванні ПрО з метою розробки програмного забезпечення (ПЗ) [2]:

1) збільшення продуктивності роботи. Традиційна технологія розробки ПЗ потребує кількох переходів від одного набору понять до іншого, що завжди пов'язано з помилками та втратою семантики ПрО. Наприклад, спочатку специфікації ПЗ повинні бути трансформовані у поняття дизайну, а потім перенесені у поняття мови програмування. Це еквівалентно розв'язанню однієї і тієї ж задачі багато разів. З інструментом DSM задача розв'язується фахівцем ПрО лише один раз на рівні абстракції, що відповідає поняттям ПрО. Немає ніякої потреби в трансформації понять, оскільки кінцевий продукт (у даному випадку, програмний код) автоматично генерується з DSM моделей;

2) краща гнучкість у відповідь на зміни. Використання для моделювання понять ПрО, а не концепцій дизайну чи програмування, приводить до можливості більш швидкого реагування на запити про зміни. При цьому зміни робляться лише у моделі ПрО на рівні понять ПрО, а програмний код для різних платформ і різновидів продукту автоматично генерується з моделі за допомогою DSM інструментів;

3) користувачі є фахівцями ПрО. Традиційною проблемою всіх команд розробників є недостатні знання о ПрО. Навіть досвідчені розробники мають часто кон-

сультуватися з експертами ПрО. В IT DSM, досвідчений експерт ПрО визначає DSL, а також способи трансформації моделей ПрО у код. Сумісно з експертом ПрО фахівець у сфері IT розроблює відповідні генератори коду. Далі користувачі, які є фахівцями ПрО, багаторазово використовують DSL для моделювання ПрО, а код автоматично генерується з DSM моделей.

Розглянемо властивості, принципи архітектури та технологію застосування існуючих інструментів предметно-орієнтованого моделювання. Зазначимо, що огляду мов моделювання та відповідних засобів присвячено кілька публікацій [3]. Наразі, найбільш зрілим інструментом для створення предметно-орієнтованих мов та підтримки процесу моделювання ПрО є MetaEdit+ [2].

Для визначення DSL у MetaEdit+ залучаються зазвичай два фахівця: експерт у ПрО та спеціаліст у сфері IT (з програмування та використання MetaEdit+). Експерт створює *метамодель*, тобто формулює поняття ПрО й правила маніпулювання цими поняттями, а фахівець у сфері IT визначає способи трансформації моделі ПрО в програмний код (за допомогою генератору коду).

Генератори коду в MetaEdit+ визначаються за допомогою мови MERL (MetaEdit+ Reporting Language), що була розроблена спеціально для трансформації графічних моделей ПрО у текстове подання. Альтернативні підходи включають доступ до моделей за допомогою SOAP (Simple Object Access Protocol), web-сервісів або запису у проміжні файли з наступною обробкою зовнішньою програмою. Зазначимо, що написання генератору програмного коду є найбільш складною частиною технології DSM, що практично неможливо здійснити без залучення фахівців у сфері IT.

Для створення предметно-орієнтованих мов моделювання MetaEdit+ надає набір інструментів *метамоделювання*, що дозволяють визначити поняття, правила маніпулювання екземплярами понять та графічні символи моделі ПрО. Побудова метамоделей у MetaEdit+ здійснюється у рамках мета-метамоделі GOPPRR (Graph-Object-Property-Port-Role-Relationship), тобто граф-об’єкт-властивість-порт-роль-відношення [2]. Це означає, що користувач будує поняття метамоделі на основі метатипів граф, об’єкт, властивість, порт, роль та відношення. Розробка моделі ПрО у MetaEdit+ здійснюється у рамках метамоделі шляхом визначення властивостей та ролей об’єктів ПрО, поєднання їх відношеннями, встановлення взаємодії об’єктів через визначені порти.

MetaEdit+ дозволяє автоматизувати розробку ПЗ, що розглядається як результат процесу моделювання ПрО. Після визначення метамоделі експертом ПрО, фахівці ПрО оперують лише предметно-орієнтованою мовою моделювання, а необхідний програмний код автоматично генерується з моделей ПрО. Генератор коду «обходить» розроблені у MetaEdit+ графові моделі ПрО, «виймає» з них інформацію і трансформує у вихідний код для визначеної цільової платформи. Із моделей, що відбивають статичні, динамічні, логічні аспекти й структуру ПрО, генератор може зробити повністю функціональний програмний код. Зазначимо, що разом з програмним кодом із моделей ПрО можливо також згенерувати технічну документацію.

Розглянемо приклад використання MetaEdit+ для створення мови моделювання для проектування цифрових годинників. На першому етапі визначення DSL здійснюється ідентифікація понять предметної області (що є близьким до побудови онтології ПрО [4]). Легко бачити, що різні моделі цифрових годинників розділяють загальне поняття кнопки, тому «кнопка» може бути обрана як поняття мови моделювання ПрО «Цифрові годинники». Іншими поняттями ПрО є дисплей, блок управління та модуль логіки. Дисплей визначає спосіб відображення інформації, блок управління - способи використання кнопок. Дисплей включає зони показу з двома цифрами, які

відбивають основні одиниці часу - секунди, хвилини та години, а також інші (службові) символи, які вказують, чи активізовані додаткові сервіси годинника (наприклад, будильник). Логічний модуль зумовлює функціональність годинника та містить визначення підпрограм, як то будильник чи секундомір.

Декомпозиція ПрО у типові поняття приводить до можливості їх багаторазового використання. Блок управління, логічний блок і модуль дисплея визначаються у візуальному редакторі MetaEdit+ як символи специфічної для ПрО діаграмної нотації, що є частиною метамоделі. Це дозволить користувачам MetaEdit+ швидко проектувати нові різновиди цифрових годинників шляхом комбінування визначених візуальних символів мови.

Після ідентифікації понять ПрО наступним кроком є їх структурування як метамоделі. Логіка годинників основана на діаграмі переходів, що є традиційним способом моделювання вбудованого ПЗ. Іншою необхідною діаграмою для моделювання годинників є структурна діаграма, що описує підпорядкованість компонентів моделі. Ці дві діаграми є своєрідними мета-метамоделями інструменту DSM. Важливо підкреслити, що ці мета-метамоделі є вбудованими у MetaEdit+ та не можуть бути змінені. Зазначимо, що як і в UML, моделі ПрО у DSM розглядаються як діаграми, а метамоделі – як певні діаграмні нотації.

Після визначення діаграмної нотації (метамоделі), створюється генератор програмного коду (зазвичай, з використанням об'єктно-орієнтованої технології). Шляхом наслідування метакласів структурної діаграми та діаграми переходів будуються предметно-орієнтовані класи генератора коду. Таким чином, практичне використання ІТ DSM (генерація коду) *потребує обов'язкового застосування певної зовнішньої мови реалізації* (зазвичай, об'єктно-орієнтованої мови програмування).

Існують також інші інструменти, призначені для побудови предметно-орієнтованих мов моделювання. Серед них зазначимо DSL Toolkit, розроблений на основі Eclipse Modeling Framework (EMF), а також JetBrains MPS. Звичайно, компанія Microsoft також не може стояти осторонь передових технологій моделювання ПрО та пропонує власний підхід. Для створення мов моделювання та засобів роботи з ними Microsoft розробила технологію та інструментарій MS DSL Tools.

Принципи інформаційної технології предметно-орієнтованого математичного моделювання

Таким чином, у розробці предметно-орієнтованої технології моделювання приймають участь великі компанії (такі як Microsoft) та консорціуми (як то Eclipse). Однак ІТ DSM не позбавлена недоліків, що пропонується подолати у рамках ІТ предметно-орієнтованого математичного моделювання (Domain-Specific Mathematical Modelling, DSMM):

1. Область застосування ІТ DSM є вузькою: інструменти DSM в основному призначені для генерації програмного коду. Визнаючи важливість такого підходу, ми розглядаємо результат процесу предметно-орієнтованого моделювання як математичну модель, до якої може бути застосовний довільний метод (окремим випадком якого є генерація даних та програмного коду).

2. Розроблені у рамках ІТ DSM мови моделювання є описовими та не дозволяють визначати методи розв'язання задач ПрО. Засоби DSMM надають можливість не лише декларативного, але й імперативного моделювання (тобто визначення способу розв'язання задачі та структури процесу обчислення).

3. В IT DSM процес моделювання розглядається у лінгвістичному аспекті, тобто як побудова синтаксичних конструкцій у рамках певної метамоделі. При цьому як в GPM, так і у DSM, моделями ПрО є діаграми, а метамоделями - діаграмні нотації. В IT DSMM лінгвістичний аспект побудови метамоделей пов'язується з математичною теорією моделювання. Для цього синтаксис метамоделі визначається на основі певного математичного апарату, а структура метамоделі включає не лише алфавіт та граматику, але і відповідні математичні операції.

4. В IT DSM відсутнє формальне визначення моделі, метамоделі та мета-метамоделі, а також методу розробки метамоделей, зокрема специфікації їх алфавітів та граматик. Крім того, є неможливим створення мета-метамоделей. В IT DSMM формально визначаються всі рівні архітектури предметно-орієнтованого моделювання, а також вводиться додатковий узагальнюючий рівень мета-мета-метамоделі, що дозоляє створювати структурно відповідні ПрО мета-метамоделі.

5. В IT DSM розірваний зв'язок між процесом моделювання ПрО фахівцем ПрО та практичним результатом застосування інструменту DSM (власне, генерацією коду). Тоді як засоби DSM призначені для використання фахівцями ПрО, розробка генераторів неможлива без залучення спеціалістів у галузі IT та застосування зовнішніх мов програмування. Інструменти IT DSMM підтримують користувача на різних етапах моделювання ПрО (починаючи з етапу висування вимог, формулювання специфікацій, архітектурного моделювання та закінчуючи генерацією коду).

6. В IT DSM неможлива зміна процесу моделювання. Інструменти DSMM надають можливість моделювання та організації процесів, що відповідають специфіці ПрО (шляхом визначення етапів процесу та умов переходу між етапами [5]).

Розробники предметно-орієнтованої технології моделювання підкреслюють семантичну протилежність IT DSM та GPM. Між тим є доцільним розгляд сильних сторін DSM та GPM з метою їх інтеграції в єдиній інформаційний технології. Для цього розглянемо більш детально методології, покладені в основу розробки мов моделювання загального призначення.

Model-Driven Engineering (MDE) можна визначити як інженерію, що породжується із моделей. До інструментів MDE в першу чергу відносять CASE-системи, що дозволяють суттєво спростити розробку ПЗ шляхом автоматичної генерації програмного коду та технічної документації із моделей ПрО. Найбільш відомим розширенням MDE є MDA (Model-Driven Architecture) - підхід, що був запропонований консорціумом OMG (Object Management Group). MDA є модельно-орієнтованим підходом до розробки архітектури ПЗ. Його сутність полягає в побудові мов моделювання (зокрема, UML) для створення моделей ПрО, а також заданні способів трансформації моделей у код мов програмування (Java, C++ та ін.).

Мови моделювання у MDA створюються за допомогою технології MOF (Meta Object Facility) [6]. Трансформація моделі у код здійснюється за допомогою *правил перетворення, що відповідають стандарту* QVT (Queries, Views, Transformations). При визначенні метамоделі MOF грає ту ж роль, що й BNF (Backus-Naur Form) при формулюванні граматик мов програмування. Таким чином, MOF можна визначити як мову для визначення метамоделей (мета-метамодель), як BNF є мовою для формулювання граматик мов програмування.

Найбільш відомим прикладом метамоделі, породженої із мета-метамоделі MOF, є уніфікована мова моделювання UML. Іншим прикладом може слугувати Common Warehouse Metamodel (CWM), що визначає метамодель (модель моделі даних) для обміну даними при використанні технології сховищ даних (*англ.* warehouse). CWM є основою для побудови протоколів обміну даних (що є екземплярами метаданих) між різним програмним забезпеченням.

MOF має 4 рівні архітектури. Верхній рівень (M3) є рівнем мета-метамоделі, що визначає мову для побудови метамоделей (що мають рівень M2). Рівень M2 (наприклад, мова UML) використовується для побудови конкретних моделей M1 (наприклад, UML-моделей). Нарешті, останнім рівнем є так званий «рівень даних» (M0), що описує конкретні екземпляри моделей ПрО рис. 1.

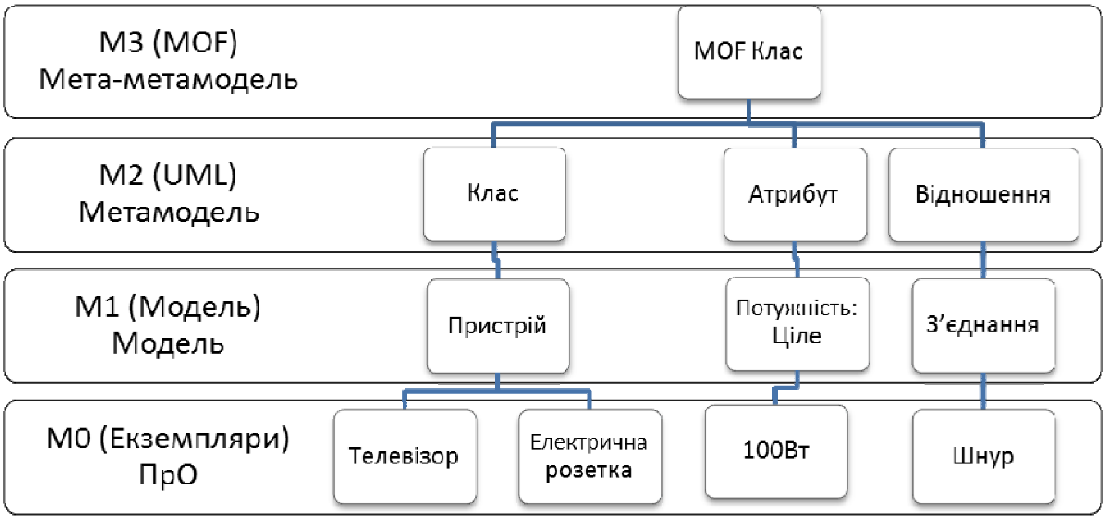


Рисунок 1 - Рівні архітектури моделювання MOF

Але зазначимо, що незважаючи на можливість побудови UML та інших мета-моделей (зокрема, CWM) у рамках мета-метамоделі MOF, такий підхід не можна віднести до предметно-орієнтованої технології моделювання. Ідея DSM полягає у розробці інструментальних засобів, що, по-перше, дозволяють *експерту* ПрО власноручно розробити предметно-орієнтовану мову моделювання та, по-друге, здійснюють підтримку процесу побудови моделі *фахівцем* ПрО.

У той же час принципи архітектури MOF (а саме, її чотирирівнева структура) були використані нами для визначення ІТ предметно-орієнтованого математичного моделювання. Особливістю ІТ DSMM є включення до метамоделей математичних операцій, що дозволяє здійснити підтримку користувача у побудові методів розв'язання специфічних для ПрО задач. Це також виключає необхідність використання додаткових до DSL мов реалізації.

ІТ DSMM узагальнює різні мета-метамоделі у рамках мета-мета-метамоделі на основі теорії множин (рис. 2), що дозволяє реалізувати перехід від загального до конкретного у процесі побудови метамоделей та їх застосування до моделювання ПрО. Зауважимо, що такий підхід не збільшує кількість рівнів архітектури моделювання ПрО. Це є можливим завдяки розгляду рівня *M2* як предметно-орієнтованої метамоделі для створення конкретних екземплярів об'єктів моделі ПрО. В ІТ DSMM найнижчим рівнем *M1* є модель ПрО, що, на відміну від MOF, має не лише описову частину - дані, але включає і процедурну складову – екземпляри методів (програмних функцій).

Наприклад, якщо розглядати як математичну основу мета-метамоделі теорію графів, то вузол та ребро графу слугують метатипами (метакласами) для породження предметно-орієнтованих типів (класів) метамоделі. (Атрибут у даному випадку розглядається як елемент вузла та ребра графу). У свою чергу поняття вузла та ребра графу узагальнюються у рамках теорії множин (що слугує рівнем M4, тобто мета-мета-метамоделлю).

Метамодель кожного рівня архітектури DSMM включає математичні операції, реалізовані у вигляді програмних функцій (API): M4 – маніпулювання з елементом множини, M3 – маніпулювання з модельними об'єктами математичного апарату (у даному прикладі – вузлом та ребром графу), M2 – визначення специфічних для ПрО методів за допомогою API рівнів M4 та M3, M1 – конкретні екземпляри методів (тобто процеси у пам'яті ЕОМ). За допомогою API також визначається граматика мов кожного рівня архітектури IT DSMM.

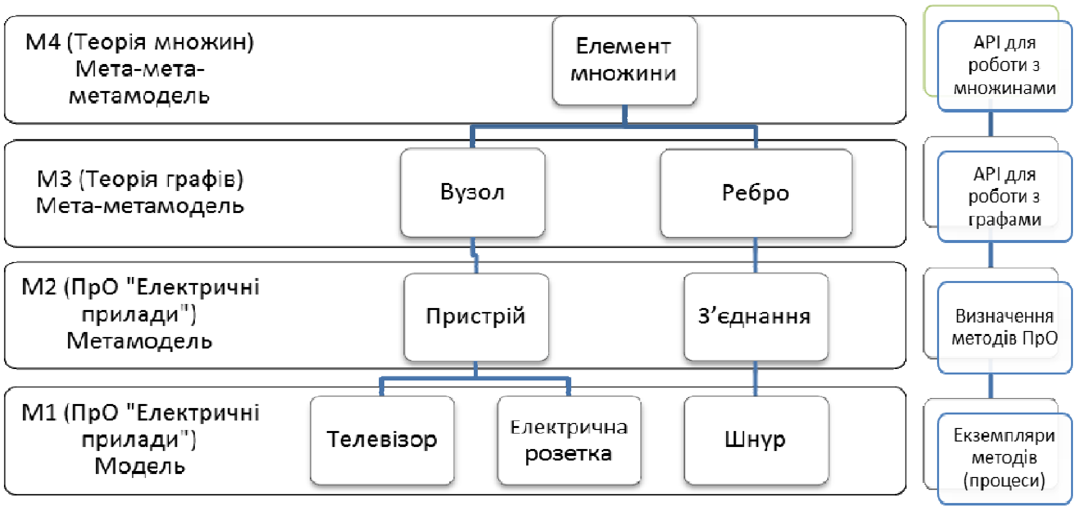


Рисунок 2 - Структурний та функціональний рівні архітектури IT та інструментів предметно-орієнтованого математичного моделювання

Висновки

Здійснений аналіз переваг та недоліків IT DSM, сутність якої полягає у створенні та застосуванні специфічних для предметних областей мов моделювання. Запропонована інформаційна технологія предметно-орієнтованого математичного моделювання, що дозволяє подолати недоліки IT DSM.

Литература

1. Межуєв В.І. Доцільність застосування UML для моделювання систем озброєння та військової техніки / Віталій Іванович Межуєв // Системи озброєння та військова техніка. – 2010. - №2 (22). – С. 122-126.
2. Steven Kelly. Domain-Specific Modeling: Enabling Full Code Generation / Steven Kelly and Juha-Pekka Tolvanen. - Wiley-IEEE Computer Society Pr. – 2008. - 427 p.
3. Межуєв В.І. Технології та інструментальні засоби предметно-орієнтованого моделювання / Віталій Іванович Межуєв // Системи управління, навігації та зв'язку. – 2010. - №3 (15). – С. 102-106.
4. Межуев В.И. Использование онтологий как моделей предметных областей / Виталий Иванович Межуев // Искусственный интеллект. – № 4. – 2009. - С. 4-11.
5. Межуев В.И. Онтологические модели систем и процесса системной инженерии / Виталий Иванович Межуев // Искусственный интеллект. - № 4. - 2010. – С. 606-616.
6. ISO/IEC 19502:2005, Information technology. - Meta Object Facility (MOF). – ANSI. - 2007. - 292 p.

Literatura

1. Mezhuev V..I. Systemy ozbroyennja ta vijs'kova tehnyka. 2010. № 2 (22). S. 122-126.
2. Steven Kelly. Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Pr. 2008. 427 p.
3. Mezhuev V.I. Systemy upravlinnja, navigacii ta zv'jazku. 2010. № 3 (15). S. 102-106.

4. Mezhuev V.I. Iskusstvennyj intellekt. № 4. 2009. S. 4-11.
5. Mezhuev V.I. Iskusstvennyj intellekt. № 4. 2010. S. 606-616.
6. ISO/IEC 19502:2005. Information technology. Meta Object Facility (MOF). – ANSI. 2007. 292 p.

RESUME

V. Mezhuyev

Information Technology of Domain-Specific Mathematical Modelling

The analysis of the information technology (IT) of Domain-Specific Modelling (DSM) is given. In spite of the power of IT DSM, its theoretical base and practical implementation have some disadvantages. The information technology of domain-specific mathematical modelling (DSMM) is proposed, allowing to overcome the lacks of IT DSM:

1. The scope of IT DSM is narrow: the DSM tools are designed for generating program code. We consider the result of the process of domain specific modelling as a mathematical model to which an arbitrary method can be applied (a special case is the generation of data and program code).

2. Developed within the IT DSM modelling languages are descriptive and do not allow to determine the methods of problem solving. DSMM enables not only declarative but imperative modelling (i.e. to determine the solving method and the process of calculations).

3. In IT DSM the process of modelling in linguistic terms is described, as the construction of syntactic structures within a metamodel. In IT DSMM the linguistic aspect of metamodels development is associated with the mathematical theory of modelling. The syntax of metamodels is based on formal apparatus, the structure of a metamodel includes not only the alphabet and grammar, but also the appropriate mathematical operations.

4. There are no formal definition of the model, metamodel and meta-metamodel and the methods of metamodels development in IT DSM. In IT DSMM the levels architecture of domain-specific modelling are formally defined, and the additional level of meta-meta-meta-model is introduced allowing to create structurally relevant meta-metamodels.

5. While IT DSM is intended for using by domain specialists, development of code generators is impossible without programmers. IT DSMM supports a user at different stages of the modelling. DSMM enables modelling and organization of specific for domain processes (by defining the steps of process and transition conditions between steps).

Стаття надійшла до редакції 21.11.2012.